HRMTime
High Resolution Timing Module

Installation and User Guide

Version 1.0

Contents

I. GETTING STARTED 5

UNPACKING THE SYSTEM AND PREPARING FOR USE 5
SAFETY CONSIDERATIONS 5
SYSTEM INSTALLATION PROCEDURES 5

II. SYSTEM OVERVIEW 6

SYSTEM CHARACTERISTICS AND SPECIFICATIONS 6
Dimensions 6
Weight 6
Power 6
Temperature 6
Specifications 6
Signal Inputs and Outputs 7
HRMTIME SYSTEM DESCRIPTION 8
Block Diagram 8
System Processor and Controller Detailed Description 9
HRMTime Typical Application 11
HRMTIME SPECIFIC FEATURES 12
Introduction 12
FIFO – TCSPC (with MACRO time) 13
FIFO – Time-Tagging 14
HISTOGRAM – TCSPC 15
HISTOGRAM – Multiscaler/Counting 16

III. SENSL INTEGRATED ENVIRONMENT (SIE) 17

INTRODUCTION 17
SYSTEM REQUIREMENTS 17
INSTALLATION 17
USING THE SENSL INTEGRATED ENVIRONMENT (SIE) 17
Main Page 17
Module Information 18
HISTOGRAM – TCSPC 19
Programmable Clock Output 19
External Clock Period 19
Reverse Plot 19
Show time bins as bars 20
Microtime LSB 20
Channel Enable and Edge Selection 20
Graphical Presentation 21
HISTOGRAM – Multiscaler/Counter 22
FIFO – TCSPC (with MACRO time) 22
Programmable Clock Output 22
Recording Length 22
Maximum Event Count 22
External Clock Period 22
Reverse Plot 22
Macro and Micro Time Configuration Selection 22
Channel Enable and Edge Selection 22
Data Recording 23
Graphical Presentation 23
Streaming TCSPC Time Tags 23
FIFO – Time Tags 26
Programmable Clock Output 26
Recording Length 27
Maximum Event Count 27
Time LSB 27
Channel Enable and Edge Selection 27
Data Recording 27
Graphical Presentation 27
Streaming FIFO Time Tags 27

Correlation 29
Target Data Set 30
Bin Size/Resolution 30
Maximum Lag 30
Graphical Presentation 30

IV. APPENDIX 31

HRMTIME REGISTERS AND LOW LEVEL DLL FUNCTIONS 31
Initialization Low Level Drivers 31
Driver - HRM_GetDLLVersion 31
Driver - HRM_SetConfigurationPath 31
Driver - HRM_RefreshConnectedModuleList 31
Driver - HRM_GetConnectedModuleCount(void) 31
Driver - HRM_GetConnectedModuleList 32
Driver - HRM_CloseModule 32
ARR – Address Route Register 33
Register Description 33
Driver - HRM_SetAddressRouteRegister 34
DRR – Data Route Register 34
Register Description 34
Driver - HRM_SetDataRouteRegister 35
LAL, LAH – Load Address LO/HI Register 35
Register Description 35
Driver - HRM_SetAddressRegister 35
LFL, LFH – Load Fill Value LO/HI Register 35
Register Description 35
Driver - HRM_SetFillValueRegister 35
UAL, UAH – Load Address LO/HI Register 35
Register Description 35
Driver - HRM_SetUSBAddressRegister 36
MBR – Mode Bits Register 36
Register Description 36
Driver - HRM_SetModeBitsRegister 37
ESR – Edge Sensitivity Register 37
Register Description 37
Driver - HRM_SetEdgeSensitivityRegister 37
RRR – Routing Reset Register 37
Register Description 37
Driver - HRM_SetRoutingResetRegister 38
MCL, MCH – Memory Count LO/HI Register 38
Register Description 38
Driver - HRM_SetMemoryCountRegister 38
FSR – Frequency Select Register 38
Register Description 38
Driver - HRM_SetFrequencySelectionRegister 38
IDR – I/O Direction Register 38
Register Description 38
Driver - HRM_SetIODirectionRegister 39
IVR – I/O Value Register 39
Register Description 39
Driver - HRM_SetIOValueRegister 39
BCL, BCH – Bin Count LO/HI Register 39
Register Description 39
Driver - HRM_SetBinCountRegister 39
UCL, UCH – USB Count HI/LO Register 40
Register Description 40
Driver - HRM_SetUSBCountRegister 40
HRS, HRMTime Status Register 40
Register Description 40
Driver - HRM_GetStatusRegister 41
PCR – Product Code Register 41
Register Description 41
Low Level Driver 41
SRR – Software Revision Register 41
Register Description 41

Low Level Driver ... 41
MIR – Module ID 1, 2, 3, 4 Register ... 41
Register Description ... 41
Driver - HRM_ModuleIDRegister ... 42
WCH – Write Count HI Register ... 42
Register Description ... 42
Driver - HRM_GetWriteCountRegister ... 42
WCL – Write Count LO Register ... 42
Register Description ... 42
Driver – HRM_GetWriteCountRegister ... 42
Non Register Specific Low Level Drivers ... 42
Driver - HRM_InitMemory ... 42
Driver - HRM_ReadMemory ... 43
Driver - HRM_ReadFIFOMemory ... 43
CORRELATION FUNCTION ALGORITHM ... 44

I. Getting Started

Unpacking the System and Preparing for Use

Contents of the package:

- HRMTime Module
- Power Supply, with country specific connector
- USB cable
- CD containing SensL Integrated Environment (SIE) software and User Guide (this document)

Unpack the contents and identify each of the components.

Safety Considerations

1. Only use the power supply supplied with the HRMTime module.
2. The power supply should be disconnected from the mains supply when the module is not in use.
3. The module is not intended for outdoor use
4. The power supply should not be opened nor should the module covers be removed at any time as there are no user adjustable components or settings, except via the SensL Integrated Environment Software.
5. Liquids should not be spilled on or into the module.

System Installation Procedures

For software driver and SensL Integrated Environment installation instructions see the SIE User Guide in Section III of this document.

II. System Overview

System Characteristics and Specifications

Dimensions

164mm (L) x 96mm (W) x 34 mm (H)

Weight

680g

Power

+5v @ 0.65 A

Temperature

Operating: 0°C to +50°C
Storage: -20°C to +70°C

Specifications

Number of channels per module	Available in 1, 2, 3 or 4 channels
Time channels per curve	1 to 4,194,304 or 1 to 8,388,608 (see Note 1)
Number of curves in memory	1 to 4,194,304 or 1 to 8,388,608 (see Note 1)
Input voltage range	LVTTL (0 - 3.3v)
Timing resolution	66ps
Minimum input pulse width	6ns
Minimum Time/Channel	27ps
Histogram/channel depth	65,535 or 4,294,967,295 bits (16 or 32 bits)
Dead time	190ns
Saturated count rate	4.5MHz
Useable count rate	9MHz (see Note 2)
Burst rate timing	Up to 100MHz (Mode dependent)
Macro Timing resolution	Down to 5ns
Memory size	8Mbytes or 16Mbytes (see Note 1)
Memory format	Dual ported linear or dual ported FIFO (mode dependent)
Readout during operation	Fully dual-ported memory (no stop start operation required)
Multi detector operation	1, 2, 3 or 4
Multi module operation	Number depends on USB capability of PC
I/O control	16 fully programmable I/O ports
Software	SensL Integrated Environment (SIE) and DLL drivers
PC Interface	High speed USB 2.0

NOTES:
1. Values dependent on memory size option
2. Useful count rate is maximum count rate without loss of greater than 50%

Signal Inputs and Outputs

Figure 1

A	Channel 0 Start Input (SMA TTL)
B	Channel 0 Stop Input (SMA TTL)
C	Channel 1 Start Input (SMA TTL)
D	Channel 1 Stop Input (SMA TTL)
E	Channel 2 Start Input (SMA TTL)
F	Channel 2 Stop Input (SMA TTL)
G	Channel 3 Start Input (SMA TTL)
H	Channel 3 Stop Input (SMA TTL)
I	USB connector
J	LEMO power supply connector (for SensL PSU use only)
K	26-way I/O port connector
L	Programmable Clock output (SMA TTL 50 ohm)

Note: The picture shows housing for a four-channel HRMTime module.

HRMTime SYSTEM DESCRIPTION

Block Diagram

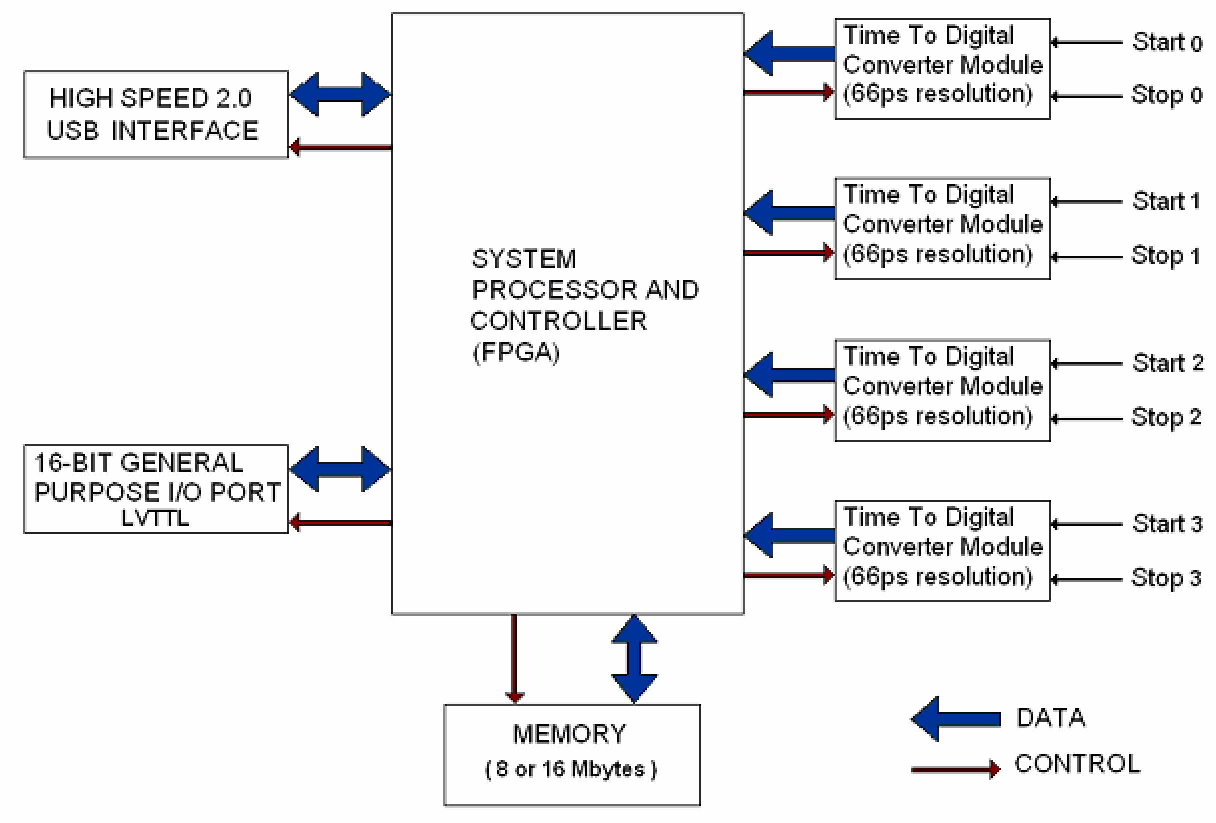


Figure 2

The HRMTime system consists of 4 ‘time to digital’ modules, 16 I/O ports, a high speed USB interface, memory storage and an FPGA based processor. The purpose of each element is explained below.

Memory
The memory module is an HRMTime format plug-in mezzanine board providing 8 or 16 Mbytes of memory.

Time to Digital Converter Module
This module is the front end of the system and is responsible for resolving the timing between the start and stop inputs of each of up to four channels. Each channel is controlled by the FPGA and can be programmed to start and stop on either LO-HI or HI-LO transitions.

High Speed USB 2.0 Interface
The USB interface is used to command/configure the HRMTime as well as download, in real-time, time-tag data to the host computer. This USB interface implements high speed USB 2.0 protocol allowing real time continuous logging of time-tag data up to rates of 4.5MHz without data loss.

16-Bit General Purpose I/O Port
This general purpose I/O port is used to allow multi-dimensional curve readings. The position of curve data, within the system memory, can be defined by these ports. These ports can be set directly by outside control lines (inputs) or by software to drive outside equipment (outputs).

System Processor and Controller

The 'System Processor/Controller' is responsible for implementing all the functionality of the HRMTime module. This module decodes commands from the USB and executes the time-bin or time-tag function accordingly. All results are saved in memory as time-bins for curve measurements or time-tags for continuous recording. In this latter mode the memory is configured as a large FIFO to allow continuous downloading of time-tag data up to rates of 4.5MHz.

System Processor and Controller Detailed Description

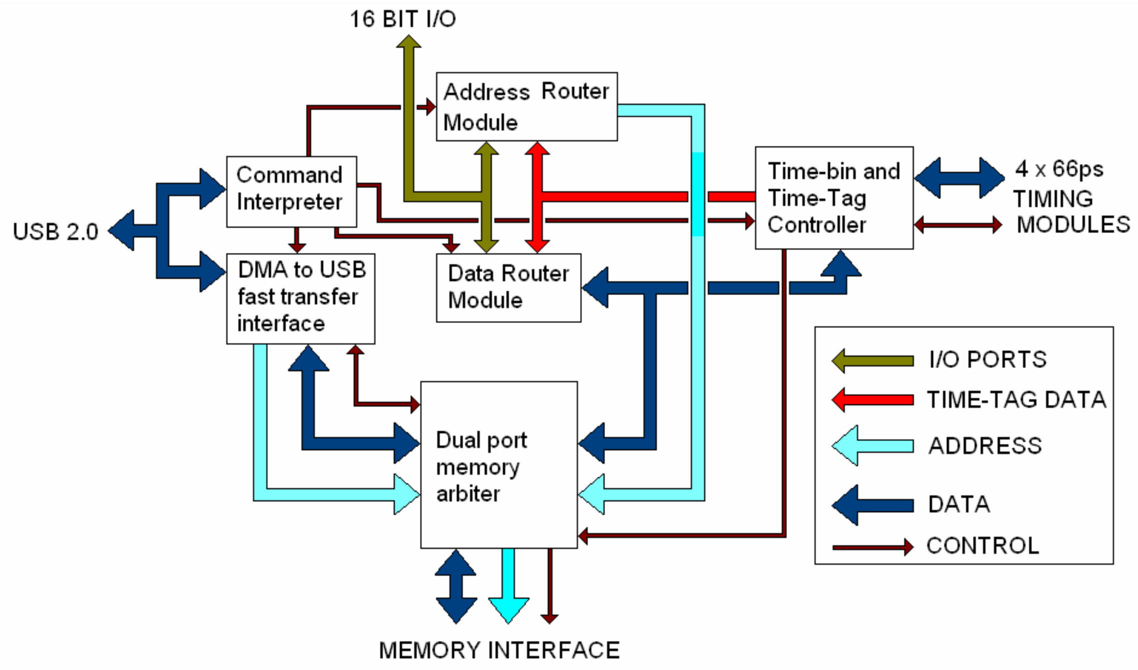


Figure 3

Command Interpreter

This module is responsible for receiving a set of commands from the host computer and controlling the system accordingly. HRMTime is a fully programmable system with a wide range of parameters that can be user defined. The **Command Interpreter** is responsible for setting these parameters and starting the execution of a particular task.

DMA to USB Fast Transfer Interface

The system memory is dual ported between the USB and the Time-Bin/Time-Tag controller. This module controls the reading of data from memory to the USB interface by means of high speed DMA block transfers. The **Command Interpreter** initializes this module with a start address and block data count. When commanded to start, this module interfaces with the **Dual Port Memory Arbiter** to read the pre-programmed data block. The rate of this process is such that data can be transferred from the memory to the USB port as fast as required. This allows the USB 2.0 high speed interface to operate at full speed without loss of data.

Time-Bin and Time-Tag Controller

This module is responsible for carrying out the particular Time-Tag or Time-Bin process as defined by the **Command Interpreter**. This module communicates with the **Timing Modules** and saves the results of the measurements in the dual ported memory. The format of these results is determined by the mode of operation. In time-bin mode, this module will use the time information from the **Timing Modules** to determine the particular bin to be incremented. In time-tag mode this module will treat the memory as a large FIFO, saving time-tag data in consecutive locations. The format of the time-tag data is determined by the **Data Router Module**.

Data Router Module

The **Data Router Module** is a complex programmable multiplexer that allows any of a wide range of inputs to be routed to any of the 32 memory data bits. In time-bin mode this module is bypassed to allow the **Time-Bin and Time-Tag Controller** to directly access the memory for the purpose of incrementing time-bins. In Time-Tag mode this module determines the format of the time-tag data. The **Command Interpreter** presets the routing of this module to define which bits of the time-tag are Time-Tag data (both Micro and Macro) from the **Time-Bin and Time-Tag Controller** and I/O data from external equipments.

Address Router Module

The **Address Router Module** is a complex programmable multiplexer that allows any of a wide range of inputs, plus an internal address counter, to be routed to any of the memory address lines. In time-tag mode this module will normally be programmed to present the internal address counter bits as the memory address. The internal address counter automatically increments after each memory write, creating a FIFO type interface. In time-bin mode the **Command Interpreter** presets the routing of this module to a mix of the address counter, time-tag data and I/O data. Routing the time-tag data to the address will create a range of consecutive bins separated by the time resolution of the LSB. The address counter bits can be used to define the base address of a particular curve whilst the I/O data can be used by external equipments to move the curve for multi-dimensional measurements.

Dual Port Memory Arbiter

This module controls the data transfers to/from system memory to the USB and Time-Bin/Time-Tag Controller. Each port presents an address, direction (R/W) and request signal. This module detects the particular request, carries out the memory access and directs the data to/from the requesting port at the requested address.

HRMTime Typical Application

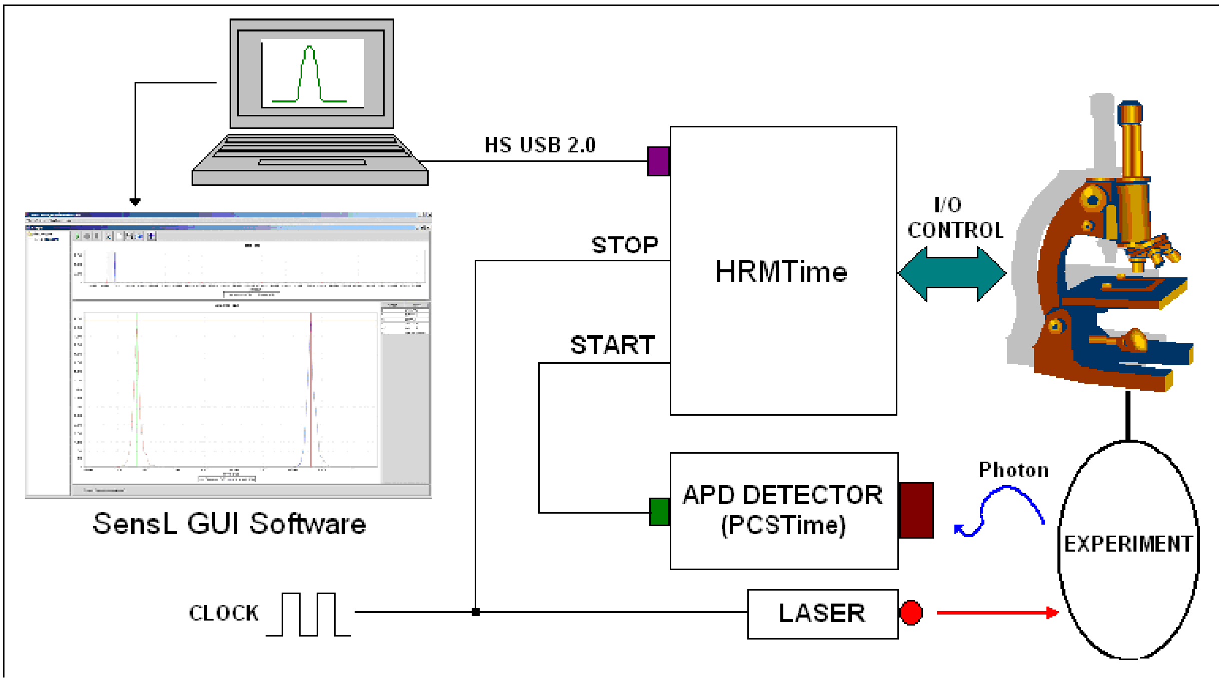


Figure 4

Note: Figure 4 shows a typical application setup utilizing a wide range of the HRMTime features. In this example the experiment is a TCSPC application where a LASER is stimulated by a clock and the time before a photon is detected is measured. The LASER is continually pulsed at a fixed frequency (typically 50MHz). The LASER output will affect a setup resulting in a photon arriving at the APD Detector such as the SensL PCSTime, PCMPlus or PCDMini. It is assumed that the rate of photons arriving at the APD is far less than the rate of the LASER pulses. As a photon is not guaranteed for each cycle of the LASER, the system will use the photon event as the start of the TCSPC process and a delayed version of the LASER pulse as the stop signal. This technique avoids countless dead cycles and simplifies the associated electronics required for recording the events.

The HRMTime module measures and records the time delay between clock and photon from the experiment and uploads the results, in real time, to the host computer via the USB interface. In some cases the experiment will involve multiple TCSPC curve measurements as the experiment changes the settings of external equipment. The programmable I/O of the HRMTime module is used to cater for such applications. The external equipment, such as a microscope, can indicate its X,Y movement to the HRMTime module allowing multiple curves to be measured. Alternatively, the HRMTime module can be programmed as outputs to control the external equipment and cause the actual X,Y positioning of the equipment.

HRMTime Specific Features

Introduction

The HRMTime module is designed to operate in a number of different modes to cater for a wide range of applications. The different operations can be split up into two major categories, HISTOGRAM and FIFO.

HISTOGRAM

Histogram modes use consecutive memory locations to store counts that represent points on a graph. These memory locations or time bins are incremented based on the value of a time measurement. For example, assuming a timing resolution of 27ps, if a time of 100ps is measured then location X is incremented. If another time of 127ps is measured then location X+1 is incremented. Hence each memory location represents a time range equal to the resolution of the timer.

Within the HISTOGRAM category there are two distinct modes of operation, TCSPC and Multiscaler/ Counter.

TCSPC

In TCSPC mode the first stop event is measured and the corresponding time bin is incremented. This is repeated to build up a histogram in memory showing the distribution of 1st events following a start input.

Multiscaler/Counter

In this mode all stop events are measured and their corresponding time bins incremented. The next following start input will reset the timer and the following events processed again. This is repeated to build up a histogram in memory showing the distribution of stop events following a start input.

FIFO

FIFO modes continually record the timing of events and save the results in consecutive locations in memory. When the last location in memory is filled, if not commanded to stop, the module continues to record data starting at the beginning of memory again. The host PC, via the USB interface, keeps up in time with the module, reading the data from memory to a file in the host computer. Hence the memory can be regarded as a very large FIFO. Providing the host PC can keep up with the module, timing data can be recorded indefinitely.

Within the FIFO category there are two distinct modes of operation, TCSPC (with MACRO time) and Time Tagging.

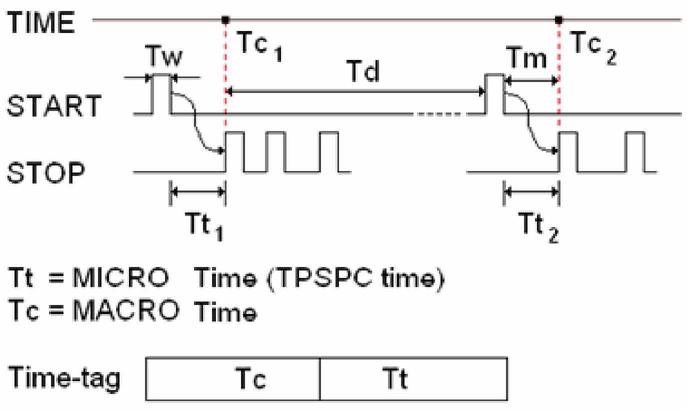
TCSPC (with MACRO time)

In this mode the module carries out the TCSPC process as described previously. However, along with the TCSPC measurement, the information stored in the FIFO also has a MACRO time that defines what time during the experiment the TCSPC measurement was made.

Time Tagging

In Time-Tagging mode the process is started with a single start pulse. The module will then fill the memory with time tags defining the time of each stop event with relation to the initial single start pulse.

FIFO – TCSPC (with MACRO time)

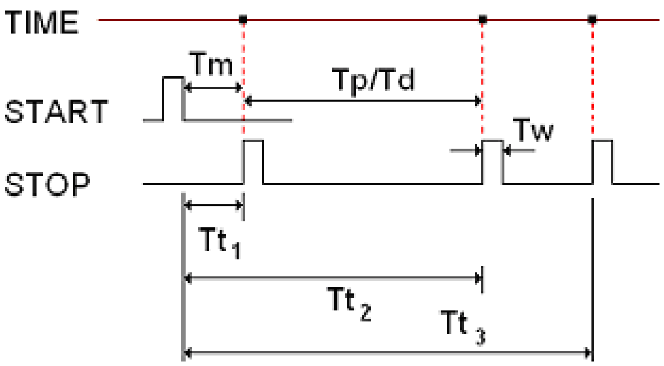


Tw	Minimum Start or Stop input pulse width	1.5ns
Td	Minimum time to process event (dead time)	190ns
Tt	MICRO Time resolution (LSB)	**27**ps
Tm	Minimim time from Start to Stop	5.2ns
Tc	MACRO Time resolution (LSB)	5ns

Figure 5

Note: In this mode the start of each channel will be the event and the stop will be a delayed version of the LASER clock. On receipt of an event the time-tag will be read and the MICRO time will be immediately reset. The reset will clear the channel ready for the next event. All subsequent stop pulses will be ignored until a new start pulse arrives. Each time-stamp will be a 32-bit word describing the TCSPC time (micro-time Tt) and the value of a free running clock defining the time within the experiment (macro-time Tc). Due to the highly flexible **Data Routing Module** the resolution and number of bits for the micro time, macro time and channel ID bits is selectable using the USB selection registers. When this process begins, 32-bit time-tags will be inserted into the shared memory. The memory will be configured as a large FIFO interfacing to the USB interface. Suitable handshake signals are implemented allowing continuous transfer of time-tags from the FIFO to the PC via the USB port. With counts of up to 4.5MHz this process can run indefinitely without loss of data.

FIFO – Time-Tagging



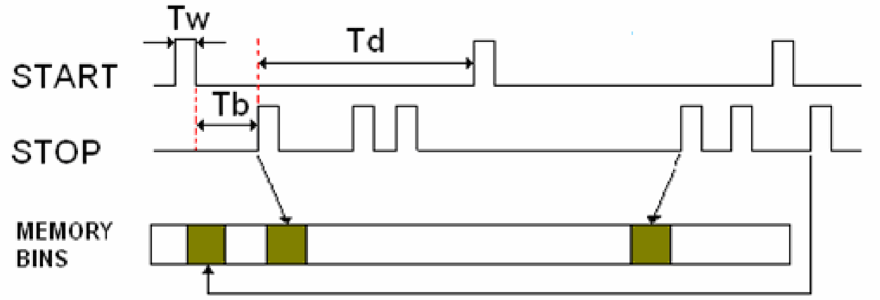
Tw	Minimum Start or Stop input pulse width	1.5ns
Tp	Minimum time between Stops (Note 1)	5.5ns
Td	Maximum average Stop rate (Note 1)	190ns
Tt	Time Tag resolution (LSB)	**27**ps
Tm	Minimim time from Start to first Stop	5.2ns

Note 1: The time to process an event (dead time) is the same as for TCSPC (190ns). However, the time measurement modules have a 256 deep FIFO allowing bursts in excess of 100MHz.

Figure 6

Note: In this mode a single start pulse will be applied to all channels. After this event all events will be time-stamped and saved to memory. Each time-tag will be comprised of two 32-bit words. These two words will provide time tag data, with a resolution of 27ps, and the channel ID. The memory will be configured as a large FIFO interfacing to the USB interface. Suitable handshake signals are implemented allowing continuous transfer of these time-tags from the FIFO to the PC via the USB port. Hence, in this mode, continuous time tagging to the host PC can be achieved indefinitely.

HISTOGRAM – TCSPC



Tw	Minimum Start or Stop input pulse width	1.5ns
Td	Minimum time to process event (dead time)	190ns
Tb	Minimum BIN size	27ps

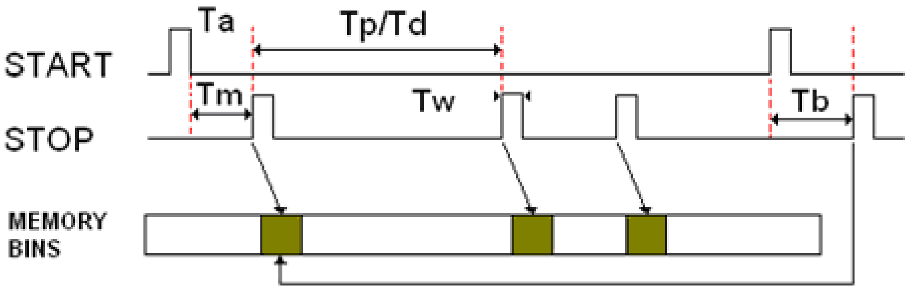
Figure 7

Note: In this mode the start of each channel will be the event and the stop will be a delayed version of the LASER clock. On receipt of an event the time-tag will be read and then the timing module will immediately be reset. The reset will clear the channel ready for the next event. Each time-stamp from the timing module will be used as an address to increment a memory location (Time-Bin). The resolution of the bins and the position of the curve in memory will be defined by the highly flexible **Address Routing Module**. The time-tag, address counter and I/O bits can all be routed to the memory address lines. This flexibility allows many TCSPC options from a simple single curve to multiple curves defined by the address counter and external control from the I/O port.

HISTOGRAM – TCSPC parameters:

Min Time Bin Size:	27ps
Max Time Bin Size:	143us
Max No. Time Bins:	8,388,608 (16 MByte memory option) 4,194,304 (8 MByte memory option)
Time Bin Depth:	65,536 or 4,294,967,296
Max Count Rate:	4.5Mcps
Max Image Size:	4096 x 4096 (16 MByte memory option) 2048 x 4096 (8 MByte memory option)

HISTOGRAM – Multiscaler/Counting



Tw	Minimum Start or Stop input pulse width	1.5ns
Tp	Minimum time between Stops (Note 1)	5.5ns
Td	Maximum average Stop rate (Note 1)	190ns
Tb	Minimum BIN size	**27**ps
Tm	Minimim time from Start to first Stop	5.2ns

Note 1: The time to process an event (dead time) is the same as for TCSPC (190ns). However, the time measurement modules have a 256 deep FIFO allowing bursts in excess of 100MHz.

Figure 8

Note: In this mode the start signal is a low frequency clock (less than 7 MHz). The stop signals will be the events. Unlike the TCSPC mode, the 27ps timing module is not reset after the first event. Due to the long clock period it will be possible for the same channel to receive a number of events per clock cycle. Hence, in this mode the time-bins will fill up to plot the occurrence of events over the period of the clock cycle. Each new start signal will reset the 27ps timing module. This allows the system to build up a plot of all the events within the start pulse cycle. Once again the flexibility of the **Address Routing Register** provides a wide range of options from single to multiple curves.

HISTOGRAM – Multiscaler/Counting Parameters:

Min Time Bin Size: 27ps

Max Time Bin Size: 143us

Max No. Time Bins: 8,388,608 (16 MByte memory option)
4,194,304 (8 MByte memory option)

Time Bin Depth: 65,536 or 4,294,967,296

Max Count Rate: 4.5Mcps

Max Image Size: 4096 x 4096 (16 MByte memory option)
2048 x 4096 (8 MByte memory option)

III. SensL Integrated Environment (SIE)

Introduction

The SIE is a user interface for setting up and controlling the HRMTime module. While the interface provides an extensive range of operating modes and measurement processes, including graphical presentation, it does not fully cover all features available in the HRMTime module.

The SIE communicates with the module via a low level DLL. This DLL has been designed to provide a set of functions that will allow full control of the HRMTime for all features. For complex experiments that require control beyond the scope of the SIE, it is expected that the user will write their own real-time application utilizing the various function in this DLL. For details of the DLL functions, see the Appendix in this document.

System Requirements

The computer used for the SensL Integrated Environment requires the following minimum configuration:

- Windows XP SP2 operating system
- 1 GByte of RAM
- At least one spare High Speed USB 2.0 port
- **.NET** Framework installed (included)
- JAVA runtime environment installed (included)
- Microsoft Visual C++ runtime components (included)

Installation

- Insert the SensL Integrated Environment installation CD.
- Use your browser to select and run **sie2installer.exe**
- Follow the installation program instructions.
- After installation is complete copy the FEATURES FILE provided for your module into the same directory as the SIE executable file.

Using the SensL Integrated Environment (SIE)

Main Page

When the SIE software is launched it will search the USB for available HRMTime modules and initialize them ready for use. Once this has been carried out the main SIE page will appear as shown in Figure 9. A list of available devices will appear. To inspect and select the various operating modes available, right click on the module name and then select the particular mode you require (see Fig 9).

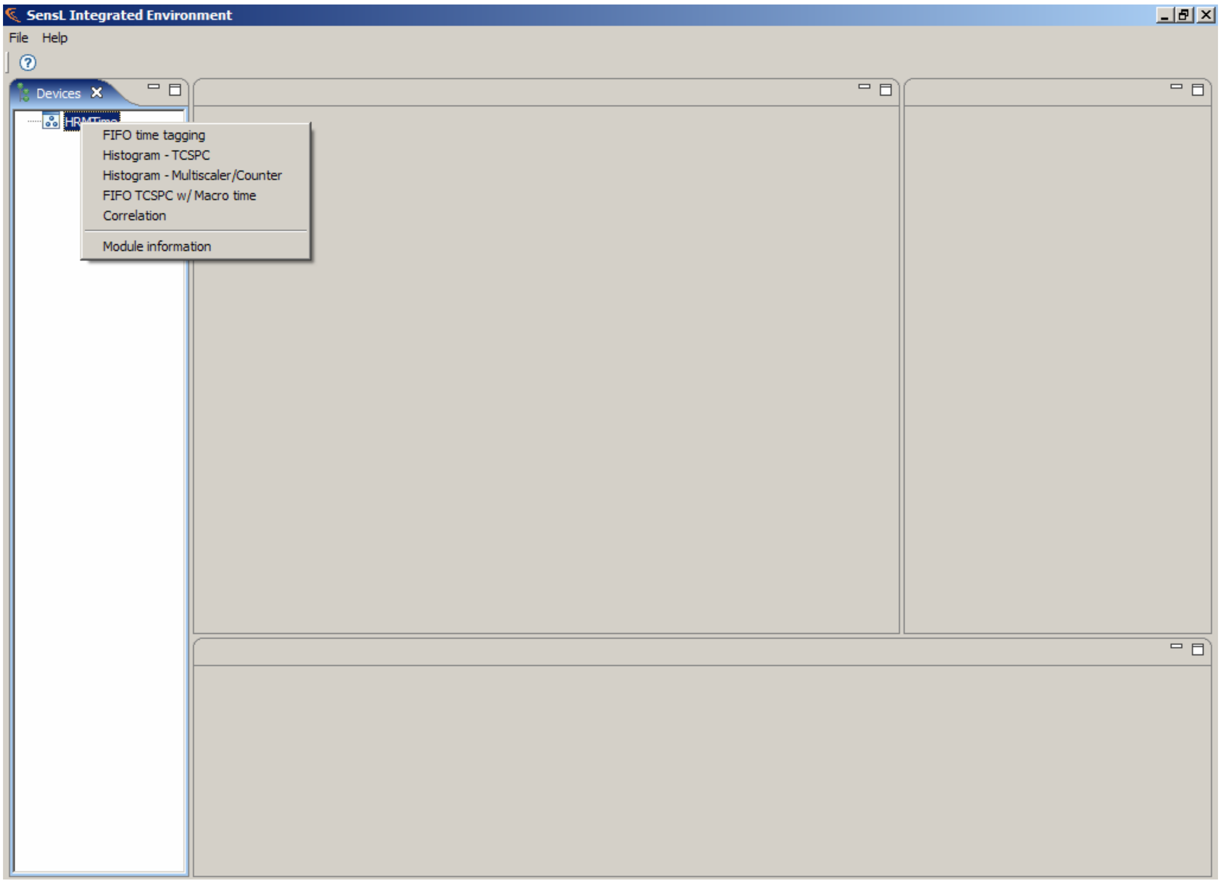


Figure 9

Module Information

This page displays the configuration information unique to this module. This information includes the module ID number, memory options and the various features enabled for the module. This page also allows the user to upgrade the internal FPGA image.

To upgrade the FPGA image the user must first click on the **Update FPGA** button. This will launch the **Update Wizard** as shown in Figure 10. Use the **Browse** button to find the RPD file for upgrading. Finally click the **Update Device** button to start the upgrade.

WARNING:
USB communication must be maintained during this process. Do **NOT** disconnect the USB cable during the update.

Updating the FPGA should only be carried out if you are instructed to do so by SensL. This procedure requires a valid RPD file as provided by SensL.

Failure to carry out this process correctly may render the module inoperable resulting in the need to return it to SensL for reconfiguration.

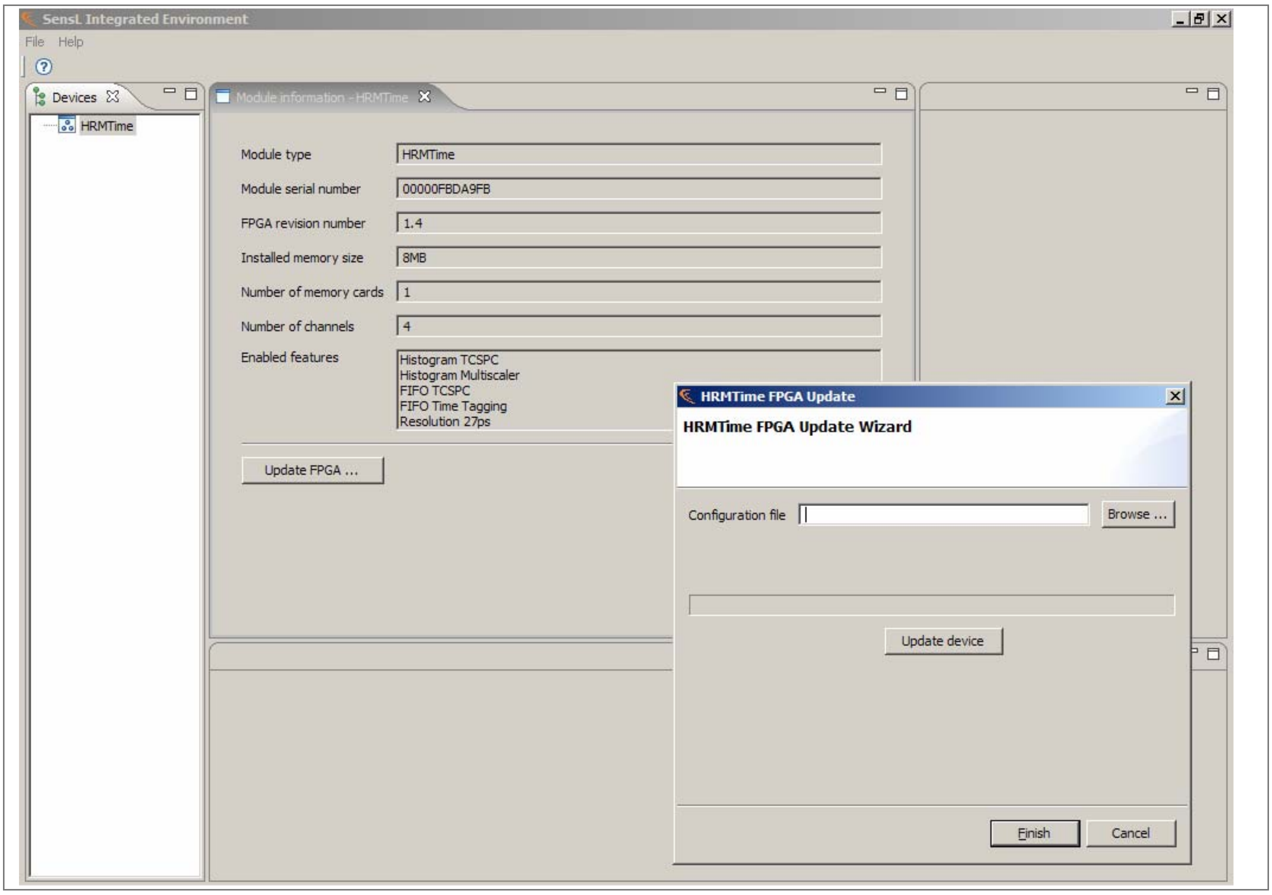


Figure 10

HISTOGRAM – TCSPC

When this page is launched the top half will display a graph page. Left click on this graph to reveal the configuration settings. The size of the configuration and graph area can be adjusted by dragging the partition to suite. Figure 11 show this page with the partition adjusted to reveal the entire configuration controls.

Programmable Clock Output

The **Programmable Clock Output** is made available for all modes and is used to set the frequency and duty cycle of the internal programmable clock. This clock is available at an SMA output for test purposes. This clock is provided for testing and diagnostics. The clock will exhibit a level of jitter that would not be suitable for accurate measurements as part of an experiment.

External Clock Period

This should be set to the period of the external LASER clock.

Reverse Plot

Due to the method of TCSPC measurement, where the start and stop events are reversed, it is sometimes useful to plot the curves with the TIME axis reversed. Selecting this option will reverse the time axis for the plot.

Show Time Bins as Bars

Selecting this option will result in the graphical output being displayed as histogram bars rather than as a single best fit line as shown in Fig 11.

Microtime LSB

The smallest bin width is 27ps. In some cases it may not be necessary to be this accurate. Selecting the LSB of the microtime defines the bin resolution. Bit 0 is the highest resolution of 27ps. Bit 1 will set the bin width to 54ps (2 x 27), bit 2 will set the bin width to 108ps (4 x 27) and so forth

The choice of bin width is application specific. Should the experiment not require such accuracy it may be better to select a lower resolution than 27ps. This will give the added advantage of allowing a wider time range over the available memory and/or more room for multiple curves.

Channel Enable and Edge Selection

These check boxes allow the individual channels to be enabled/disabled and the sensitivity of the START/STOP inputs to be specified.

Note: Press the **Apply changes** button to set your selected configuration.

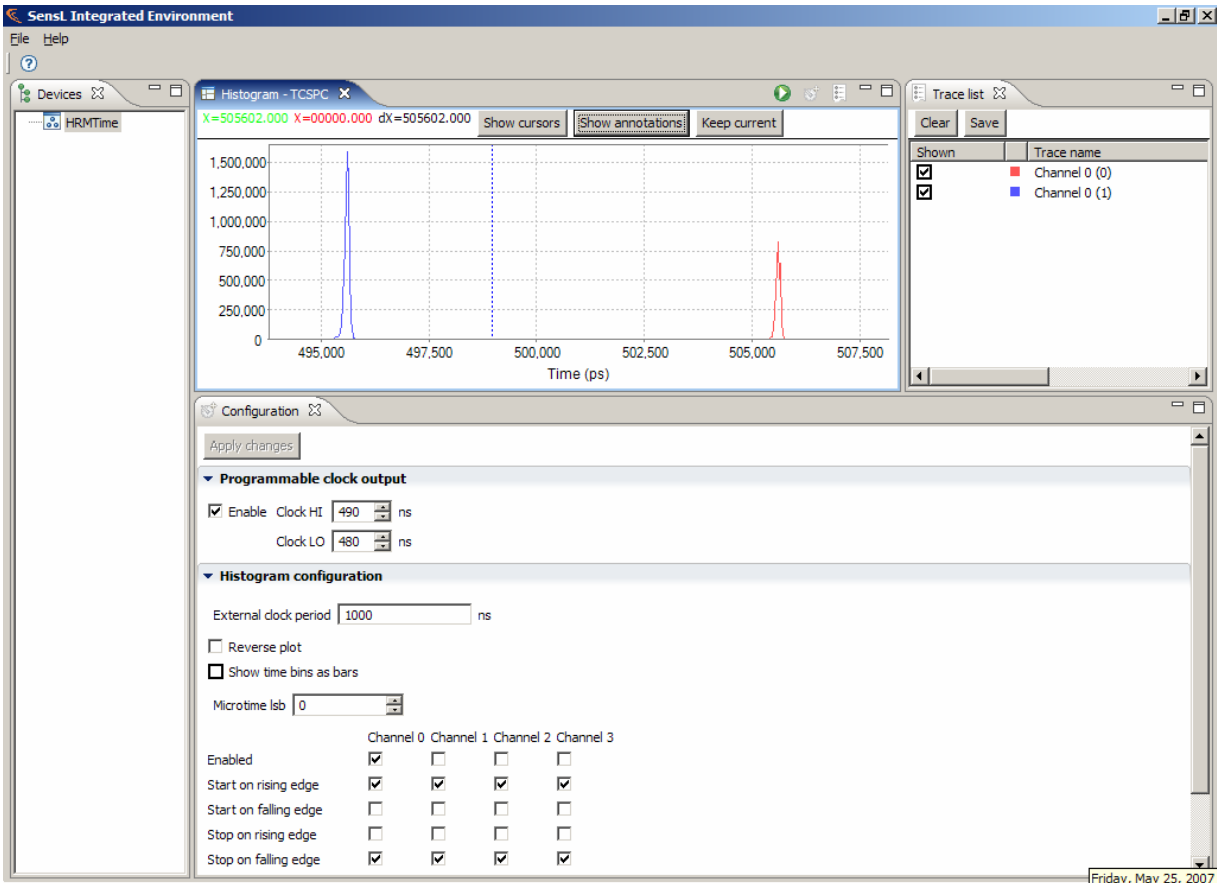


Figure 11

Graphical Presentation

Once the configuration is selected, the configuration page can be removed by clicking on the **X** tab to display the graph section only (see Figure 12).

To start processing, click on the green right arrow at the top of the page. This can be done numerous times to display multiple traces on the same graph. Fig 11 shows the result of a simple TCSPC experiment. The right hand side of the display shows the traces. In this case two traces are displayed both from channel0. These traces can be saved to file or cleared by selecting them with the relevant checkbox and using the save and clear buttons.

Once the processing has been stopped, the graph can be analyzed.

To zoom in, hold right mouse button down and size selection box over area of interest.

To zoom out, sweep mouse from right to left with right mouse button held down.

To measure between points, click on the graph with right button and left button to position the two cursors. The position of these two cursors and their X and Y differences will be displayed at the top of the graph (see Fig 12).

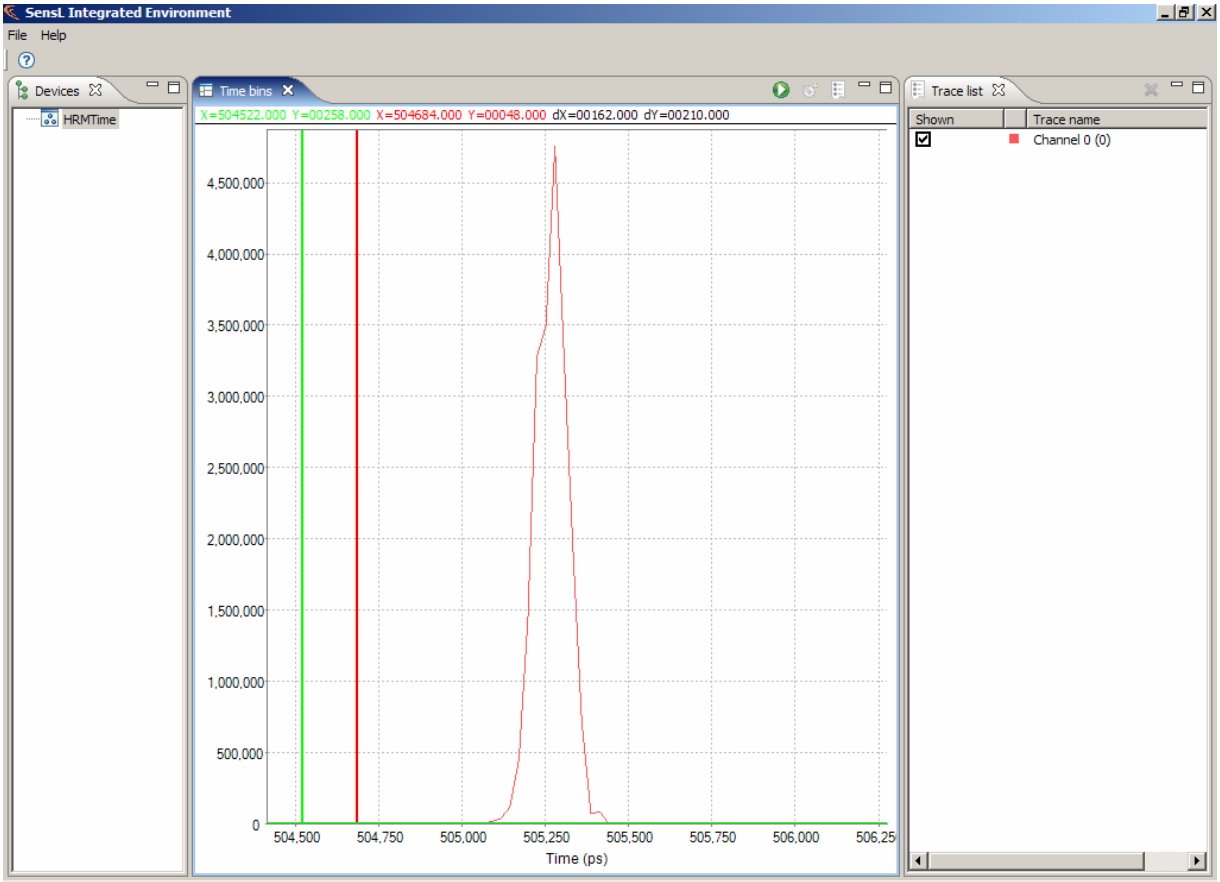


Figure 12

HISTOGRAM – Multiscaler/Counter

When in this mode the configuration setup and operation is identical to the **HISTOGRAM-TCSPC** mode. In TCSPC mode the system repeatedly plots the time of the first event after a LASER pulse. In **Multiscaler/Counter** mode the operation is very different. The start event is a slow clock of less than 7MHz. The system records and saves all stop events in their respective time bins. Each start event will reset the timer and a new set of stop events will be added to the existing array of time bins. This process will result in a histogram being built up of all the events following the start. This is particularly useful for plotting pulse shapes, decay curves etc.

FIFO – TCSPC (with MACRO time)

When this page is launched the top half will display a graph page. Left click on this graph to reveal the configuration settings. The size of the configuration and graph area can be adjusted by dragging the partition to suite.

This page is a graphical demonstration of the TCSPC with MACRO time feature of the HRMTime module. The HRMTime allows the user to carry out TCSPC and save time tags. These time tags consist of the TCSPC measurement plus a MACRO time defining at what time during the experiment the measurement was made. For normal operation, this feature would use data streaming that would allow the user to continually record time-tags indefinitely to a PC file. This mode of the SIE records for a given period or until the HRMTime memory is full.

Programmable Clock Output

The Programmable Clock Output is made available for all modes and is used to set the frequency and duty cycle of the internal programmable clock. This clock is available at an SMA output for test purposes. This clock is provided mainly for testing and diagnostics. The clock will exhibit a level of jitter that would not be suitable for accurate measurements as part of an experiment.

Recording Length

The recording length should be set to the desired period over which the TCSPC measurements are to be taken. Note that the processing will stop prematurely if the **Maximum Event Count** is reached.

Maximum Event Count

This defines the maximum number of events to be stored before recording stops. This value is used to ensure the storage data size does not exceed the capacity of the system.

External Clock Period

This should be set to the period of the external LASER clock.

Reverse Plot

Due to the method of TCSPC measurement, where the start and stop events are reversed, it is sometimes useful to plot the curves with the TIME axis reversed. Selecting this option will reverse the time axis for the plot.

Macro and Micro Time Configuration Selection

These fields allow the user to select the number of MACRO and MICRO bits and resolution to appear in the 32-bit time tag word. As bit counts and resolution are changed the resulting roll over times and resolution values will be automatically displayed in the boxes to the right hand side. If the user attempts to input an illegal value the relevant text boxes will turn red.

Channel Enable and Edge Selection

These check boxes allow the individual channels to be enabled/disabled and the START/STOP inputs sensitivity to be specified.

Note: Press **Apply changes** button to set your selected configuration.

Data Recording

Once the configuration is selected, the mode is now ready for recoding data. In **FIFO – TCSPC** mode two forms of data recording are available, Graphical Presentation and Streaming TCSPC Time Tags

Graphical Presentation

In this mode recording is carried out at the module until the recording length or the maximum event count is reached. To start processing, click on the green right arrow at the top of the page. Figure 13 shows the result of a simple experiment. The right hand side of the display will show which channels are active. In this example only channel 0 is active.

After starting the experiment the module will run for the Recording Length or until the memory is full. Once the process has stopped the top graph will display a plot of the event frequency over time. The user can now use the cursors to select a time period of the top graph. When this is done the software will automatically plot the TCSPC curve for the time tags over that particular period.

The user can, at any time, save these curves and run the experiment again. Fig 13 shows an example of this mode of operation. The user can, for each run, save a particular TCSPC curve or set of curves and compare it with other curves at different MACRO time ranges and/or runs.

Once the processing has been stopped, the graphs can be analyzed.

To zoom in, hold right mouse button down and size selection box over area of interest.

To zoom out, sweep mouse from right o left with right mouse button held down.

To measure between points click on graph with right button and left button to position two cursors. The position of these two cursors and their X and Y differences will be displayed at the top of the graph (see Fig 13).

This mode of operation is particularly useful for carrying out preliminary tests to determine the best configuration, before carrying out a full experiment using continuous streaming of results to a PC file.

Streaming TCSPC Time Tags

Once the configuration is selected, the user can now select a path and file name for saving the data. It is recommended that the file name have a suffix of CSV. Doing this will allow the file to be easily viewed using a spreadsheet package. Once this is done the process can be started by clicking the **Start** button. Clicking the **Start** button will start the module streaming all time tags to the chosen file until the recording time is reached, the maximum event count is reached or the process is manually stopped.

After the process is stopped the file will be available for viewing. Figure 14 shows a section of a typical output file.

As can be seen in Fig 14, four columns are used to define the tag number, channel ID, macro time and micro time.

The example experiment used the test waveform set to 1MHz with a 50% duty cycle. This signal was fed directly into both the start and stop inputs of channel 0. The start was set to trigger on the LO-HI transition and the stop was set o trigger on the HI-LO transition. As can be seen in Fig 14, the time-tags are repeating every 200 counts of the MACRO time. As the resolution of the MACRO time was set to 5ns, this represents a repetition rate of 1us (200 x 5 ns). The TCSPC time is typically 18712. As the resolution of the MICRO time was set to 27ps, this represents a TCSPC MICRO time of ~505ns (18712 x 27ps). The value of 505ns rather than the theoretical value of 500ns is due to the jitter of the clock and the quality of the cabling.

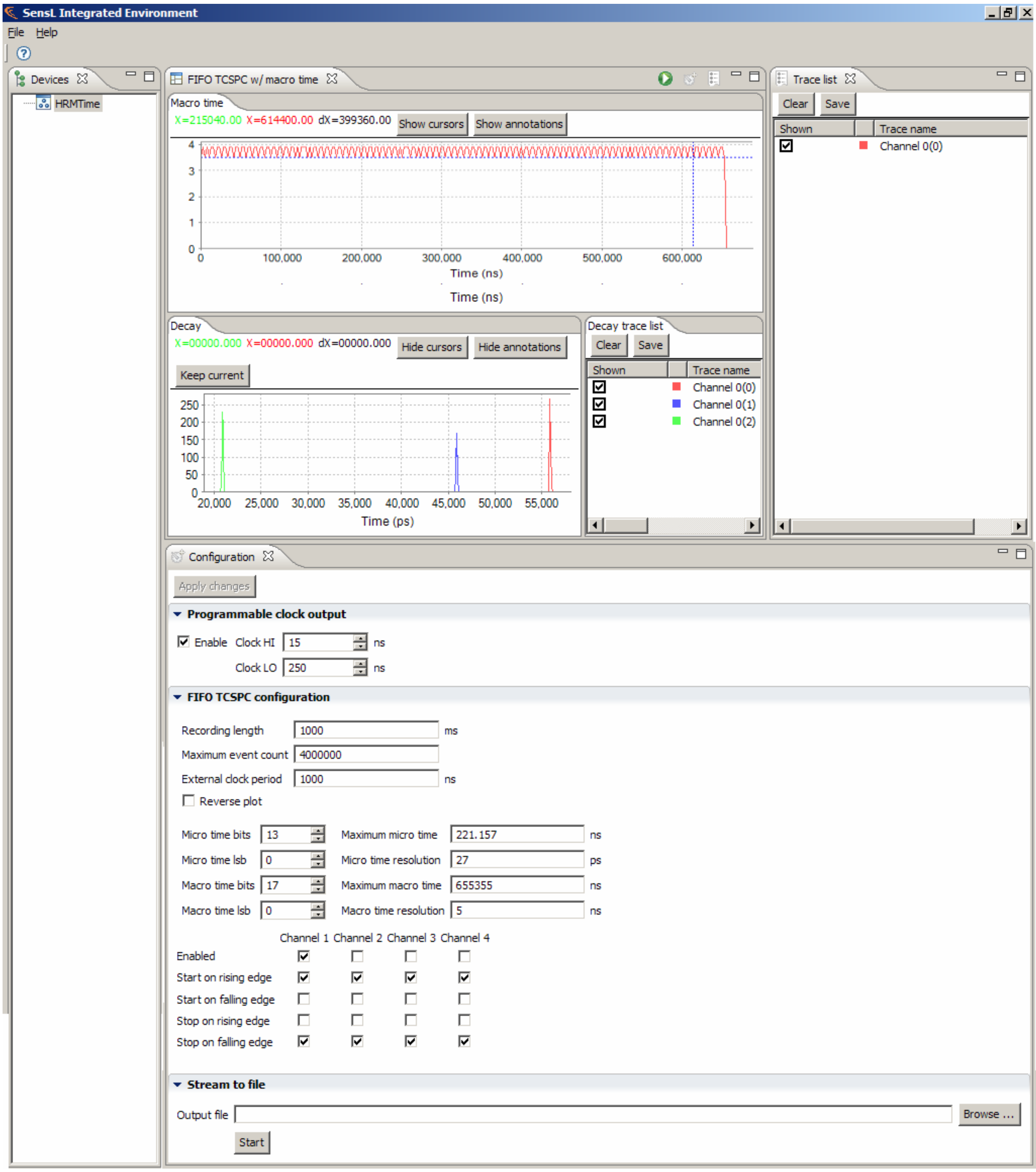
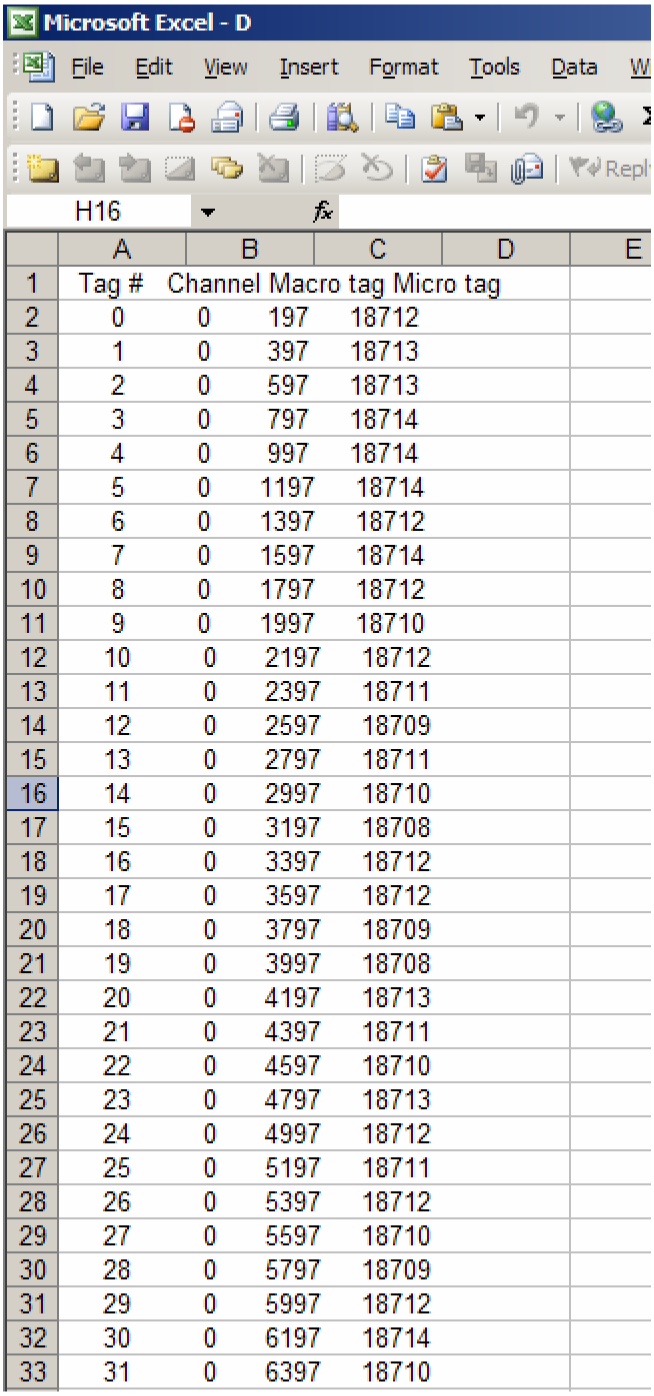

SensL Integrated Environment
File Help
Devices
HRMTime
FIFO TCSPC w/ macro time
Macro time
X=215040.00 X=614400.00 dX=399360.00
Show cursors
Show annotations
Time (ns)
Decay
X=00000.000 X=00000.000 dX=00000.000
Hide cursors
Hide annotations
Keep current
Time (ps)
Decay trace list
Clear
Save
Shown
Trace name
Channel 0(0)
Channel 0(1)
Channel 0(2)
Trace list
Configuration
Apply changes
Programmable clock output
Enable Clock HI 15 ns
Clock LO 250 ns
FIFO TCSPC configuration
Recording length 1000 ms
Maximum event count 4000000
External clock period 1000 ns
Reverse plot
Micro time bits 13
Maximum micro time 221.157 ns
Micro time lsb 0
Micro time resolution 27 ps
Macro time bits 17
Maximum macro time 655355 ns
Macro time lsb 0
Macro time resolution 5 ns
Channel 1 Channel 2 Channel 3 Channel 4
Enabled
Start on rising edge
Start on falling edge
Stop on rising edge
Stop on falling edge
Stream to file
Output file
Browse ...
Start

Figure 13



Tag #	Channel	Macro tag	Micro tag
0	0	197	18712
1	0	397	18713
2	0	597	18713
3	0	797	18714
4	0	997	18714
5	0	1197	18714
6	0	1397	18712
7	0	1597	18714
8	0	1797	18712
9	0	1997	18710
10	0	2197	18712
11	0	2397	18711
12	0	2597	18709
13	0	2797	18711
14	0	2997	18710
15	0	3197	18708
16	0	3397	18712
17	0	3597	18712
18	0	3797	18709
19	0	3997	18708
20	0	4197	18713
21	0	4397	18711
22	0	4597	18710
23	0	4797	18713
24	0	4997	18712
25	0	5197	18711
26	0	5397	18712
27	0	5597	18710
28	0	5797	18709
29	0	5997	18712
30	0	6197	18714
31	0	6397	18710

Figure 14

FIFO – Time Tags

This page is used to continuously stream time tags of events after a singe start pulse is received. Once the configuration is selected, the user can start the module recording. The module will wait for the first single start pulse and then start recording every stop event until the recording time is reached, the maximum event count is reached or the process is manually stopped. In this mode there is no TCSPC. The first start pulse will begin recording. All following stop events will be stamped and saved. Any following start pulses during the process will be ignored. Hence, all events, after the initial start signal, can be saved indefinitely.

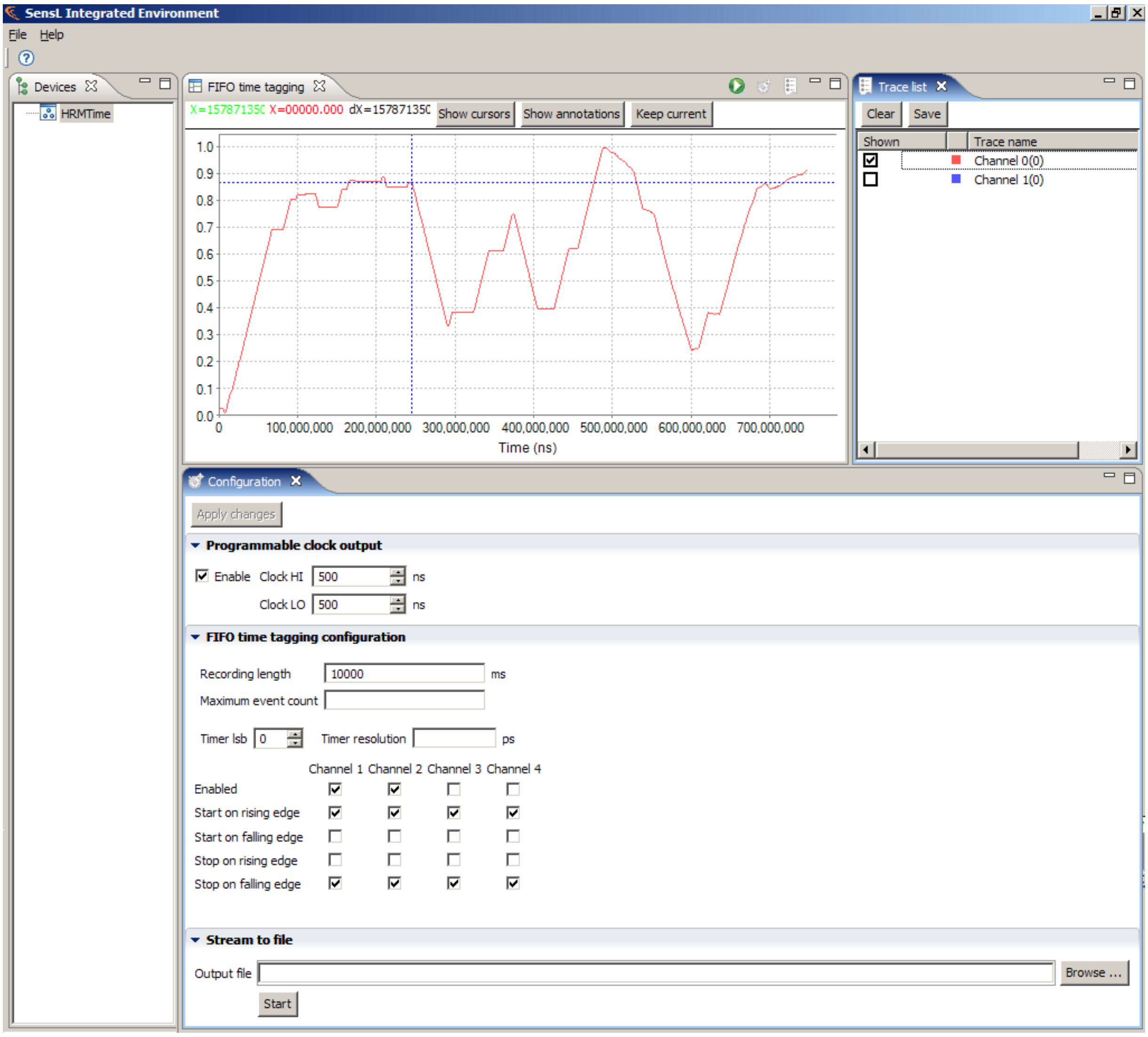


Figure 15

Programmable Clock Output

The Programmable Clock Output is made available for all modes and is used to set the frequency and duty cycle of the internal programmable clock. This clock is available an SMA output for test purposes. This clock is provided for testing and diagnostics only. The clock will exhibit a level of jitter that would not be suitable for accurate measurements as part of an experiment.

Recording Length

The recording length should be set to the desired period over which the TCSPC measurements are to be taken.

Maximum Event Count

This defines the maximum number of events to be stored before recording stops. This value is used to ensure the storage data size does not exceed the capacity of the system.

Time LSB

This is used to select the resolution of the time tag. Selecting bit 0 will give the time tag a resolution of 27ps. Bit 1 will set the resolution to 54ps (2 x 27), bit 2 will set the resolution to 108ps (4 x 27) and so forth.

Channel Enable and Edge Selection

These check boxes allow the individual channels to be enabled/disabled and the START/STOP inputs sensitivity to be specified.

Note: Press the **Apply changes** button to set your selected configuration.

Data Recording

Once the configuration is selected, the mode is now ready for recoding data. In **FIFO – Time Tagging** mode two forms of data recording are available, Graphical Presentation and Streaming FIFO Time Tags.

Graphical Presentation

In this mode recording is carried out at the module until the recording length or the maximum event count is reached. To start processing, click on the green right arrow at the top of the page. The right hand side of the display will show which channels are active. On completion the graph will display a plot of event density (frequency) over time.

This mode of operation is particularly useful for carrying out preliminary tests to determine the best configuration, before carrying out a full experiment using continuous streaming of results to a PC file.

Streaming FIFO Time Tags

Once the configuration is selected, the user can now select a path and file name for saving the data. It is recommended that the file name have a suffix of CSV. Doing this will allow the file to be easily viewed using a spreadsheet package. Once this is done the process can be started by clicking the **Start** button. Clicking the **Start** button will start the module streaming all time tags to the chosen file until the recording time is reached, the maximum event count is reached or the process is manually stopped. In this mode there is no TCSPC. The first start pulse will begin recording. All following stop events will be stamped and saved in the target PC file. Any following start pulses during the process will be ignored. Hence, all events, after the initial start signal, can be saved to file indefinitely.

After the process is stopped the file will be available for viewing. Figure 16 shows a section of a typical output file.

As can be seen in Fig 16, four columns are used to define the tag number, channel ID, macro time and micro time.

The time tag for this mode consists of 2 x 32-bit words. The first word is a micro time that has a resolution down to 27ps. This timer will roll over at the count of 5308415 (Hex 50FFFF). The MACRO counter is a count of how many times the MICRO counter has rolled over.

Note: The LSB value of 27ps is not an exact value. This value is a simplified (rounded up) value that is suitable for all other modes. The true LSB value is 26.9851ps. As this mode involves the continuous running of the MICRO clock for very long periods it is recommended that the value of 26.9851 be used to avoid a cumulative error occurring over long periods of time. All other bit precisions should be calculated using this value. Therefore the precision of bit 1 is 53.9702ps (2 x 26.9851) and so on.

Microsoft Excel - D

	A	B	C	D
1	Tag #	Channel	Macrotime	Microtime
2	0	0	0	18714
3	1	0	0	55771
4	2	0	0	92825
5	3	0	0	129877
6	4	0	0	166931
7	5	0	0	203985
8	6	0	0	241040
9	7	0	0	278093
10	8	0	0	315146
11	9	0	0	352200
12	10	0	0	389252
13	11	0	0	426302
14	12	0	0	463358
15	13	0	0	500412
16	14	0	0	537464
17	15	0	0	574515
18	16	0	0	611567
19	17	0	0	648620
20	18	0	0	685669
21	19	0	0	722725
22	20	0	0	759776
23	21	0	0	796829
24	22	0	0	833879
25	23	0	0	870932
26	24	0	0	907984
27	25	0	0	945040
28	26	0	0	982089
29	27	0	0	1019144
30	28	0	0	1056196
31	29	0	0	1093249
32	30	0	0	1130301
33	31	0	0	1167356

Figure 16

Example: If the MACRO time value is 28 and the MICRO time is 11232 with a resolution of 27ps (26.9851ps) then the absolute time of the tag from the start pulse is:

((28 * 5308416) + 11232) x 26.9851ps = 4011250921ps. **Note:** 5308416 = Hex 510000.

The example experiment used the test waveform set to 1MHz with a 50% duty cycle. This signal was fed directly into both the start and stop inputs of channel 0. The start was set to trigger on the LO-HI transition and the stop was set o trigger on the HI-LO transition. As can be seen in Fig 16, the MICRO times are typically increasing by ~37054 counts. With a resolution of 26.9851ps this represents a time tag every ~1us.

Correlation

The Correlation feature allows the user to carry out cross and auto correlation on FIFO-TCSPC streams for both the TCSPC values and the MACRO times.

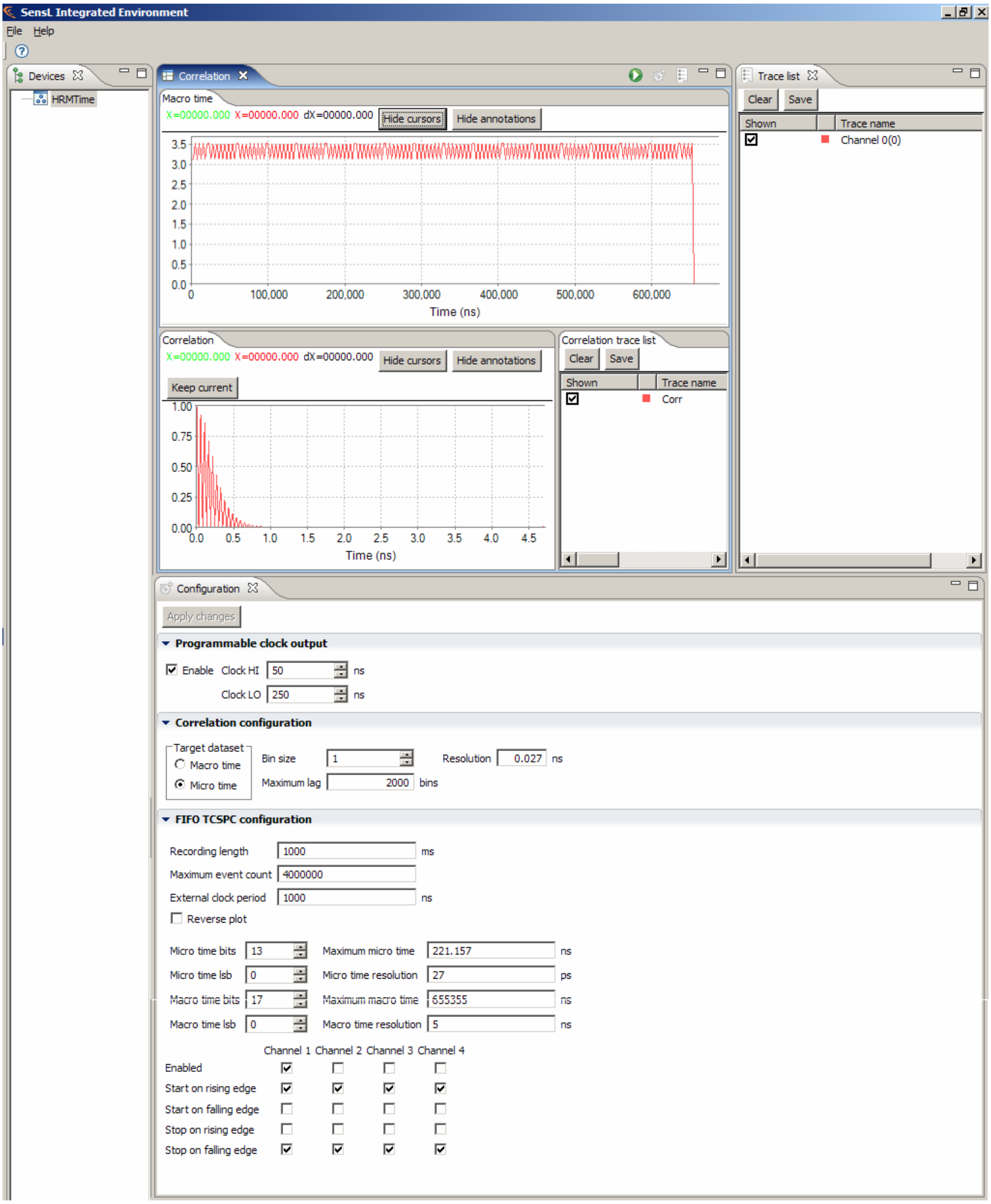


Figure 17

In **Correlation** mode the configuration setup is identical to **FIFO-TCSPC** mode. However, channel select is restricted to a maximum of two channels. A single channel selected will result in auto-correlation on that input. Two channels will result in cross-correlation on the two channels. Further correlation specific settings are as follows:

Target Data Set

Use these two mutually exclusive radio buttons to select correlation on the MICRO (TCSPC) or MACRO time.

Bin Size/Resolution

This setting determines the bin size to be used for the correlation function. Increasing this value will direct the correlation function to group greater numbers of consecutive time tags into software bins. These bins are then used for phase sweeping the streams to create the correlation curve.

Maximum Lag

This defines the maximum number of bins to be used for carrying out the correlation algorithm.

Graphical Presentation

Once the configuration is selected, the mode is now ready for recording data. In this mode recording is carried out at the module until the recording length or the maximum event count is reached. To start processing, click on the green right arrow at the top of the page.

After starting the experiment the module will run for the Recording Length, the maximum event count is reached or until the memory is full. Once the process has stopped the top graph will display a plot of the event frequency over time. The bottom graph will display the correlation curve as specified by the configuration parameters. For details of the correlation algorithm see the Appendix in this document.

IV. Appendix

HRMTime Registers and Low Level DLL Functions

The control and setup of the HRMTime is carried out by a series of commands to a set of configuration registers within the module. To simplify the control of these registers, a set of low level drivers, in a DLL, is available. The low level drivers will return an **HRM_STATUS** of value **HRM_OK** or **HRM_ERROR**.

Initialization Low Level Drivers

Before the user can read/write to these configuration registers communication must be established with the module. To do this the following low level driver functions must be used.

Driver - HRM_GetDLLVersion

HRM_API const char* WINAPI HRM_GetDLLVersion(void)

This function returns a pointer to a text string describing the revision level of the drivers.

Driver - HRM_SetConfigurationPath

HRM_API void WINAPI HRM_SetConfigurationPath(char* path)

path: Pointer to text string defining path.

This function is used to define the path where the configuration data for the module resides.

Driver - HRM_RefreshConnectedModuleList

HRM_API bool HRM_RefreshConnectedModuleList(void)

This function can be called at any time to determine if the list of connected modules has changed. This can be used to periodically poll the USB bus to determine if modules have been connected or disconnected.

Driver - HRM_GetConnectedModuleCount(void)

HRM_API UINT WINAPI HRM_GetConnectedModuleCount(void)

This function is used to determine how many HRMTime modules are currently connected to the USB bus.

Driver - HRM_GetConnectedModuleList

HRM_API void WINAPI HRM_GetConnectedModuleList(HANDLE* handleList)
handleList: Pointer to array of HRMTime handles for initialization.

This function initializes an array of HRMTime handles to allow communication with all HRMTime modules present on the USB bus. The size of the array must be greater or equal to the number of modules detected using the function **HRM_GetConnectedModuleCount**.

Driver - HRM_CloseModule

HRM_STATUS WINAPI HRM_CloseModule(HANDLE handle)
On completion of the application, this function must be called to release the handle and close the session.

Example:

```
Int             moduleCount;
HANDLE          handleArr[20];

HRM_RefreshConnectedModuleList();

moduleCount = HRM_GetConnectedModuleCount();

if(moduleCount)
{
        HRM_GetConnectedModuleList(handleArr);

        APPLICATION CODE HERE
}
else
   printf("No HRMTime modules detected");
```

In this example the **APPLICATION CODE** can address up to **'moduleCount'** HRMTime modules.

Now that communication with the module has been established the configuration registers can be programmed using the associated low level driver.

ARR – Address Route Register

Register Description

The method of time-binning is based on using the received time-tag data and discrete I/O inputs to form the address in memory for time-bin processing. In its simplest form, a time-tag could be used as the address bus so that each time-bin is separated by the resolution of the least significant bit. On receipt of a time-tag the system outputs the time-tag as an address and then increments that location (time-bin). In the HRMTime system further data bits are included in the address selection to allow multiple curve plotting based on multiple channel inputs and discrete inputs for X,Y array plotting. To allow maximum flexibility the AAR register can define any bit to be placed in any position within the address bus for the shared memory.

Understanding the ARR is critical as it is the controller that defines the resolution, curve count and array size of all measurements.

Before programming the ARR the user must first assert an **RRR** (Route Register Reset) command to initialize the system. Once this is done the ARR is then programmed by sending 27 consecutive writes. The address bus of the memory is 27 bits (A26-A0). Starting with A0, each write defines the bit number of the 'Address Option Bits' to be routed to that particular address bit. The 'Address Option Bits' are as follows:

AOB[24.. 0]	TagData
AOB[51..25]	Address Counter
AOB[67..52]	16-bit I/O Data
AOB[79..68]	Pixel Counter
AOB[91..80]	Line Counter
AOB[92]	Fix to logic 0

TagData

This is the time-tag data as received by the Pico-Second Timing Interface. Bits 23,24 define on which channel the time-tag was received – 00, 01, 10 or 11. The bits 22 down to 0 define the time with bit 0 being the LSB (LSB = 26.9851ps).

Address Counter

These bits provide a 27 bit counter that can be routed to the address bus. This counter can be pre-loaded with a given value. After each write to memory this counter will be automatically incremented. This counter would be most used when the system is in time-tag mode. Here the system reads time-tags and stores them in consecutive locations in memory. These bits are also available to be used in Time-bin mode. However, in this case the address is not incremented. Instead the address counter bits are used purely as an offset address in memory for saving curves.

16-Bit I/O Data

The value of the 16 I/O data bits can be routed to any address line. This would be useful for plotting X, Y curves. For example, the I/O could be used as 8-bit X and 8-bit Y inputs allowing a 256 x 256 array of curves to be plotted.

Pixel Counter/Line Counter

The 16-bit I/O method of X,Y plotting is limited to 256 x 256 arrays. An alternative method that allows larger arrays is to use 2 discrete inputs to clock counters that in turn can be used as the address in memory. The HRMTime module provides a 12-bit Pixel Counter and a 12-bit Line Counter. The Pixel Counter is incremented by clock inputs to IODATA(0) and the line counter is incremented by clock inputs to IODATA(1). If these bits are routed to the address lines then the user can command the HRMTime

module to move from one curve to the next by clocking the IODAT(0) and IOADAT(1) lines. This would allow arrays of up to 4096 x 4096.

Fix to logic 0
Selecting this bit will drive the particular address line low. This is used for driving the chip select line of a single memory card. If two memory cards are used then the chip select should be an address counter bit.

Example:

WRITE: 1,2,3,4,5,6,7,8, 23,24, 25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41

This would set up the system recording 4 curves, one from each channel. Each curve would consist of 256 bins (8 bits) with a bin size of 60ps. This reduced resolution is due to bit 1 (second bit) of the time-tag being routed to address bit 0. The channel bits 23, 24 will move each channel event to a different curve. The base start address of these curves will be defined by the pre-programmed value of the **Address Counter**.

Driver - HRM_SetAddressRouteRegister

HRM_STATUS WINAPI HRM_SetAddressRouteRegister(HANDLE handle, BYTE* arrData)

handle:	HRMTime module handle
arrData:	Array of bytes to write to the address route register.

DRR – Data Route Register

Register Description

When the HRMTime system is in time-tag mode it will continually save time-stamps to memory. Each time-stamp will always be 32-bits, however the format of the time-stamp is programmable using the DRR. To allow maximum flexibility the DRR register can define any 'Data Option Bit' bit to be placed in any position within the 32-bit time-tag.

Before programming the DRR the user must first assert an RRR (Route Register Reset) command to initialize the system. Once this is done the DRR is then programmed by sending 32 consecutive writes. Starting with D0, each write defines the bit number of the 'Data Option Bits' to be routed to that particular data bit within the time-tag. The 'Data Option Bits' are as follows:

DOB[24.. 0]	TagData
DOB[56..25]	Macro Counter
DOB[57]	Fix to logic 0

TagData
This is the time-tag data as received by the Pico-Second Timing Interface. Bits 23,24 define the channel the time-tag was received on – 00, 01, 10 or 11. The bits 22 down to 0 define the time with bit 0 being the LSB (LSB = 26.9851ps).

Macro Counter
When time-tag recording the user may, along with the TCSPC time, wish to record the chronological time that the event occurred. A 32-bit Macro Time Counter is made available that is cleared at the start of time-tag processing and will increment every 5ns. The user can select a range of these bits to provide a macro time to time-stamp each time-tag.

Example:

WRITE: 0,1,2,3,4,5,6,7 26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47 23,24

This would set up the time-tag as follows:

D[7..0]	=	TCSPC time (LSB = 26.9851ps)
D[29..8]	=	22-bit Macro time with LSB resolution of 5ns
D[31..30]	=	2-bit channel code 00, 01, 10, 11 for channels 0 to 3.

Driver - HRM_SetDataRouteRegister

HRM_STATUS WINAPI HRM_SetDataRouteRegister(HANDLE handle, BYTE* drrData)

handle: HRMTime module handle

drrData: Array of bytes to write to the data route register.

LAL, LAH – Load Address LO/HI Register

Register Description

These two registers are used to initialize the **'Address Counter'** (see ARR register) to a pre-defined value. The order of loading the initialization address **must** be LAL followed by LAH. The LAL command will define the least significant 16 bits (A15 down to A0) of the counter. The least significant 11 bits of the LAH command will define counter bits A26 down to A16. On completion of the LAH command the '**Address Register'** will be loaded with the new value.

Driver - HRM_SetAddressRegister

HRM_STATUS WINAPI HRM_SetAddressRegister(HANDLE handle, ULONG arData)

handle: HRMTime module handle

arData: 32-bit address to set LAH, LAL to.

LFL, LFH – Load Fill Value LO/HI Register

Register Description

The user can command the HRMTime module to fill a range of memory with a given value. The value used for this command is defined using these 2 commands. The initialization value is a 32-bit value. The most significant 16 bits is defined by LFH and the least significant 16 bits is defined bits LFL.

Driver - HRM_SetFillValueRegister

HRM_STATUS WINAPI HRM_SetFillValueRegister(HANDLE handle, ULONG fvrData)

handle: HRMTime module handle

arData: 32-bit value to set LFH, LFL to.

UAL, UAH – Load Address LO/HI Register

Register Description

These two registers are used to initialize the USB address counter. The block DMA transfers from memory to the USB start at the address defined by these two commands. On completion of each USB transfer the USB address counter is automatically incremented. This address is a 32-bit **'long word'** address. All USB block transfers are carried out in long words (4 bytes at a time). The order of loading the initialization address **must** be UAL followed by UAH. The UAL command will define the least significant 16 bits (A15 down to A0) of the counter. The least significant 10 bits of the UAH command will

define counter bits A25 down to A16. On completion of the UAH command the 'Address Register' will be loaded with the new value.

UAL:

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

UAH:

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Nu	Nu	Nu	Nu	Nu	Nu	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16

Driver - HRM_SetUSBAddressRegister

HRM_STATUS WINAPI HRM_SetUSBAddressRegister(HANDLE handle, ULONG uarData)
handle: HRMTime module handle
uarData: 32-bit address to set UAH, UAL to.

MBR – Mode Bits Register

Register Description

This register defines a number of settings for the HRMTime module as follows:

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Rmd	Mem	Nu	Nu	Nu	Nu	Nu	Nu	Nu	BCe	Rvd	Rvd	Size	Md2	Md1	Md0

Md[2..0] These bits define in which mode the HRMTime module will operate.

000 = Fill 'n' memory locations with the **LFL,LFH** value.

The value of 'n' is defined by the **MCL,MCH** registers.
The start address is defined by the **LAL,LAH** registers.

001 = Run in TIME-TAG with TCSPC mode.
010 = Run in TIME-TAG continuous mode.
011 = Run in TIME-BIN with TCSPC mode.
100 = Run in TIME-BIN continuous mode.

All other Md[] combinations are reserved.
Power-up default = 000.

Size
In TIME-BIN mode the bin size can be set for 16 bits or 32 bits. If **'Size'** is set to '1' the bin size will be 16 bits. With **'Size'** set to '0' the bin size will be 32 bits. Note that when the size is 16 bits the memory address defined by the ARR (Address Route Register) is a 16-bit word address. In all other cases the ARR shall define a 32-bit long words address.
Power-up default = 1

Rvd
Bits reserved. **Must be set to '1'.**

BCe
With this bit set to '1' the BINCNT (BCH, BCL) feature is enabled. If this bit is '0' then the effect of the BCH and BCL is disabled.

Mem
This bit, when set to '1', will start the high speed USB memory or time-tag transfer processor. Taking this bit to '0', at any time, will immediately put the processor into reset.

Rmd
This bit, when set to '1' will start the mode processor. The mode of operation is defined by the Md bits. Taking this bit to '0', at any time, will immediately put the processor into reset.

Driver - HRM_SetModeBitsRegister

HRM_STATUS WINAPI HRM_SetModeBitsRegister(HANDLE handle, USHORT mbrData)

handle: HRMTime module handle

mbrData: 16-bit value to write to the MBR.

ESR – Edge Sensitivity Register

Register Description

All start and stop inputs can be programmed to produce an event on either a +Ve or -Ve transition. This register defines the edge sensitivity for each input as follows:

D15	*D14*	*D13*	*D12*	*D11*	*D10*	*D09*	*D08*	*D07*	*D06*	*D05*	*D04*	*D03*	*D02*	*D01*	*D00*
FN3	**FP3**	**SN3**	**SP3**	**FN2**	**FP2**	**SN2**	**SP2**	**FN1**	**FP1**	**SN1**	**SP1**	**FN0**	**FP0**	**SN0**	**SP0**

SP3, SP2, SP1, SP0 = Set to '1' for start event of corresponding channel on positive edge.
SN3, SN2, SN1, SN0 = Set to '1' for start event of corresponding channel on negative edge.
FP3, FP2, FP1, FP0 = Set to '1' for end event of corresponding channel on positive edge.
FN3, FN2, FN1, FN0 = Set to '1' for end event of corresponding channel on negative edge.

Note:
The positive and negative settings must never both be set to '1'. Only one edge is allowed.
To disable an input, set both the negative and positive bits to '0'.

Driver - HRM_SetEdgeSensitivityRegister

HRM_STATUS WINAPI HRM_SetEdgeSensitivityRegister(HANDLE handle, USHORT esrData)

handle: HRMTime module handle

esrData: 16-bit value to write to the ESR.

RRR – Routing Reset Register

Register Description

A write to this register:
Resets the ARR and DRR registers ready for programming
Clears the WCH and WCL registers
Clears the Memory Wrap Error bit in the status register

Driver - HRM_SetRoutingResetRegister

HRM_STATUS WINAPI HRM_SetRoutingResetRegister(HANDLE handle, USHORT rrrData)

handle: HRMTime module handle

rrrData: Don't care

MCL, MCH – Memory Count LO/HI Register

Register Description

When the system is commanded to initialize the memory, these registers will define the block size of 32-bit memory locations to be written to with the value defined by the **Load Fill Value** registers. When the system is put into Time-Tag or TCSPC Time-Tag mode, these registers will define the number of 32-bit memory locations to be stored as time-tag data before halting. If these registers are set to 0 the time-tag processor will run until commanded to stop by resetting the state machine.

For further details **see MBR** (Md bits = 000).

Driver - HRM_SetMemoryCountRegister

HRM_STATUS WINAPI HRM_SetMemoryCountRegister(HANDLE handle, ULONG mcrData)

handle: HRMTime module handle

mcrData: Number of 32-bit locations to process

FSR – Frequency Select Register

Register Description

In certain experiments it may be required for the host computer to control the 16-bit I/O as outputs. These outputs can be used to control the movement of external equipment such as a microscope. In this case it may be desirable to use an internally generated clock for both the LASER and the stop inputs for the TCSPC process. The HRMTime module provides a programmable frequency output that can be used for this purpose. This register defines the number of 5ns cycles required to complete the HI and LO parts of the cycle. The most significant 8 bits of the FSR defines the HI time and the least significant 8 bits defines the LO time. However, there is an offset of 1 such that:

Setting this value to 0x0000 would result in an output waveform 5ns low followed by 5ns high.
Setting this value to 0x0309 would result in an output waveform 20ns HI, 50ns LO.

Driver - HRM_SetFrequencySelectionRegister

HRM_STATUS WINAPI HRM_SetFrequencySelectionRegister(HANDLE handle,USHORT fsrData)

handle: HRMTime module handle

fsrData: Value to write to the FSR

IDR – I/O Direction Register

Register Description

The 16-bit I/O signals of the HRMTime can be programmed to be inputs or outputs. The value of this register defines the direction of each I/O bit. Setting a bit in this register to '1' will program the corresponding I/O bit as an output. Setting a bit in this register to '0' will program the corresponding I/O bit as an input.

Driver - HRM_SetIODirectionRegister

HRM_STATUS WINAPI HRM_SetIODirectionRegister(HANDLE handle, USHORT iodrData)

handle: HRMTime module handle

iodrData: Value to write to the IDR

IVR – I/O Value Register

Register Description

Writing to this register sets any I/O bit, enabled as an output, to the value of its corresponding bit.

Driver - HRM_SetIOValueRegister

HRM_STATUS WINAPI HRM_SetIOValueRegister(HANDLE handle, USHORT iovrData)

handle: HRMTime module handle

iovrData: Value to write to the IVR

BCL, BCH – Bin Count LO/HI Register

Register Description

These registers define the maximum number bins that can occur during the period of the TCSPC clock. This count can be calculated as:

Bin Count = MOD(Clock Period/Resolution) + 1

Resolution = 26.9851ps.

The Bin Count is a 23-bit number. BCL defines the least significant 16 bits and BCH defines the most significant 7 bits. It is **important** that this value is correctly set for both Time-Binning **and** Time-Tagging.

Note: This feature is disabled if bit 6 of the mode register is clear.

Driver - HRM_SetBinCountRegister

HRM_STATUS WINAPI HRM_SetBinCountRegister(HANDLE handle, ULONG bcrData)

handle: HRMTime module handle

bcrData: 32-bit value to write to the BCH, BCL registers

UCL, UCH – USB Count HI/LO Register

Register Description

These registers define the number of 32-bit words to be read from the USB high speed interface.

Driver - HRM_SetUSBCountRegister

HRM_STATUS WINAPI HRM_SetUSBCountRegister(HANDLE handle, ULONG ucrData)

handle: HRMTime module handle

ucrData: Number of 32-bit words to be read from the USB high-speed interface

HRS, HRMTime Status Register

Register Description

Reading this register will report the status of the HRMTime module as follows:

D15	*D14*	*D13*	*D12*	*D11*	*D10*	*D09*	*D08*	*D07*	*D06*	*D05*	*D04*	*D03*	*D02*	*D01*	*D00*
MC	**MT2**	**MT1**	**MT0**	**CH1**	**CH0**	**0**	**0**	**0**	**0**	**0**	**OV**	**ME**	**FC**	**HS**	**TP**

TP: If set to '1' , the TCSPC time-tag/time-bin processor is active.
HS: If set to '1', the memory high-speed data transfer processor to the USB port will be active.
FC: If set to '1' the FPGA configuration processor is active.
ME: If set to '1' indicates that a TIME-TAG memory WRAP-ROUND error has occurred.
OV: This bit will be cleared when the state machine is reset.
OV will go to '1' if, during processing, the 32-bit 5ns resolution macro timer wraps-round from maximum back to 0.

CH: These two bits define the module type 1, 2 or 4 channels.

CH1	*CH0*	*TYPE*
0	0	1 Channel
0	1	2 Channel
1	0	3 Channel
1	1	4 Channel

MT: These three bits define the memory card type/size being used:

MT2	*MT1*	*MT0*	*SIZE*
0	0	0	Reserved
0	0	1	Reserved
0	1	0	Reserved
0	1	1	Reserved
1	0	0	32 Mbytes
1	0	1	16 Mbytes
1	1	0	8 Mbytes
1	1	1	No card installed

MC: If set to '1', only one card of **MT** type is installed. If '0', two cards of **MT** type are installed.

Driver - HRM_GetStatusRegister

HRM_STATUS WINAPI HRM_GetStatusRegister(HANDLE handle, USHORT *srData)
handle: HRMTime module handle
srData: Pointer for saving current 16-bit HRS value

PCR – Product Code Register

Register Description

Reading from this register will report the PRODUCT code. The LS byte will report the product ID and the MS byte defines any variants from the standard product. For a standard HRMTime module this value should read 0x0001.

Low Level Driver

HRM_STATUS WINAPI HRM_GetProductCodeRegister(HANDLE handle, USHORT *pcrData)
handle: HRMTime module handle
pcrData: Pointer for saving current 16-bit PCR value

SRR – Software Revision Register

Register Description

Reading from this register will report the current rev of the FPGA code.

Low Level Driver

HRM_STATUS WINAPI HRM_GetSoftwareRevisionRegister(HANDLE handle, USHORT *srrData)
handle: HRMTime module handle
srrData: Pointer for saving current 16-bit SRR value

MIR – Module ID 1, 2, 3, 4 Register

Register Description

Reading from these four registers will report the contents of the 'on-board' serial ID chip. The contents of the serial ID chip is comprised of 64 bits. The MSB of **ID-1** will be the first bit returned from the ID chip. The LSB of **ID-4** will be the last bit returned from the ID chip.

Driver - HRM_ModuleIDRegister

HRM_STATUS WINAPI HRM_ModuleIDRegister(HANDLE handle, BYTE *midData)

handle: HRMTime module handle

midData: Pointer for saving text string of the HRMTime module ID

WCH – Write Count HI Register

Register Description

When operating in time-tag mode, this register will contain the number of 1K (1024 bytes) blocks of data that have been written to memory by the time-tag processor. When the time-tag processor is running, this register should be used to track the memory for continuous download of data.

Note: This register automatically wraps around at the maximum address as defined by the memory configuration bits in the status register.

Driver - HRM_GetWriteCountRegister

HRM_STATUS WINAPI HRM_GetWriteCountRegister(HANDLE handle, ULONG *wrrData)

handle: HRMTime module handle

wrrData: Pointer for saving current 32-bit value of WCH and WCL registers

WCL – Write Count LO Register

Register Description

When operating in time-tag mode, this register will contain the residual bytes (0-1023 bytes) that have been written to memory by the time-tag processor. The value of the WCH and WCL are not locked. The WCH should be used for tracking the memory data. Once the time-tagging has been stopped the WCL register should be used to download any remaining data.

Driver – HRM_GetWriteCountRegister

See WCH register

Non Register Specific Low Level Drivers

Driver - HRM_InitMemory

**HRM_STATUS WINAPI HRM_InitMemory(HANDLE handle,
ULONG addr, ULONG len,
ULONG fillData)**

Fill a block of memory with a specific bit pattern.

handle:	HRMTime module handle
addr:	32-bit starting address
len:	Number of 32-bit locations to fill
fillData:	32-bit value to fill the memory with

Driver - HRM_ReadMemory

**HRM_STATUS WINAPI HRM_ReadMemory(HANDLE handle,
USHORT modeMask,
ULONG addr,
ULONG len,
BYTE *buf)**

Read a block of data from a given location in memory.

handle:	HRMTime module handle
modeMask:	Mask to define desired state of mode register bits when executing the function
addr:	32-bit starting address
len:	Number of 32-bit locations to read
buf:	Pointer to buffer for storing the data

Driver - HRM_ReadFIFOMemory

**HRM_STATUS WINAPI HRM_ReadFIFOMemory(HANDLE handle,
USHORT modeMask,
ULONG addr,
ULONG len,
BYTE *buf)**

Read a block of data from a given location in memory when card is operating in FIFO mode.

handle:	HRMTime module handle
modeMask:	Mask to define desired state of mode register bits when executing the function
addr:	32-bit starting address
len:	Number of 32-bit locations to read
buf:	Pointer to buffer for storing the data

Correlation Function Algorithm

In **correlation** mode the system will carry out a single sweep of software bins with all the bins initially set to 0. On completion the system will calculate the correlation between two inputs (cross correlation) or correlation on a single input (auto correlation) and save the result. The number of time-bins to be used for the calculation and the resolution of the bin is programmable. The results will comprise of a number of values equal to the number of time-bins used for the calculation. The position of each value represents the level of phase shift between the input streams. The first value corresponds to a shift of 0, the second a shift of 'T' etc, where 'T' is the resolution of the time-bin. The value of each result represents the level of correlation at that particular phase.

Cross/ Auto correlation is a standard method of estimating the degree to which two series are correlated. Consider two series x(i) and y(i) where i=0,1,2...N-1. The cross correlation r at delay d is defined as

$$r = \frac{\sum_i [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i-d) - my)^2}}$$

Where **mx** and **my** are the means of the corresponding series. If the above is computed for all delays d=0,1,2,...N-1 then it results in a cross correlation series of twice the length as the original series.

$$r(d) = \frac{\sum_i [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i-d) - my)^2}}$$

There is the issue of what to do when the index into the series is less than 0 or greater than or equal to the number of points. (i-d < 0 or i-d >= N) The most common approaches are to either ignore these points or assuming the series x and y are zero for i < 0 and i >= N. In many signal processing applications the series is assumed to be circular in which case the out of range indexes are "wrapped" back within range, ie: x(-1) = x(N-1), x(N+5) = x(5) etc

The range of delays d and thus the length of the cross correlation series can be less than N, for example the aim may be to test correlation at short delays only. The denominator in the expression above serves to normalize the correlation coefficients such that -1 <= r(d) <= 1, the bounds indicating maximum correlation and 0 indicating no correlation. A high negative correlation indicates a high correlation but of the inverse of one of the series.